36th EUROPEAN ROTORCRAFT FORUM

Session: Systems

AUTONOMOUS OBSTACLE AVOIDANCE FOR ROTORCRAFT INCLUDING VEHICLE CONSTRAINTS AND MISSION OBJECTIVES

by

J. Mooney, M. Costello

School of Aerospace Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

SEPTEMBER 7-9, 2010

PARIS

FRANCE

Autonomous Obstacle Avoidance for Rotorcraft Including Vehicle Constraints and Mission Objectives

John G. Mooney[†], **Mark F. Costello**[*]
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA

Nomenclature

A, B, C: Continuous system matrices
A_d, B_d: Discrete system matrices
b: Sensor bias
E: Vector of error between commanded and predicted output over the prediction horizon
Hp: Prediction horizon (number of time steps)
k_{CA}, k_{CAB}: Model predictive control 'gain' matrices
Q_1, Q_2: Cost function weighting matrices
$\vec{u}$: Control vector
U: Vector of velocity commands over the prediction horizon
$v_{x,c}, v_{y,c}, v_{z,c}$: Commanded velocities in x, y,
V_{max}: Maximum aircraft velocity
w: Sensor noise
$\vec{x}, \dot{\vec{x}}$: State vector
$\tilde{x}_{k+i}$: Predicted state at time step i
$\vec{y}$: Output vector
$\tilde{Y}$: Vector of predicted outputs
Y_c: Vector of commanded outputs
Δs: Integration step size in arclength
ϕ: Potential field
σ: Standard deviation of sensor noise
τ_x, τ_x, τ_x: Time constant of first order lag response to velocity commands and z directions

Introduction

Military applications of unmanned rotorcraft often require an aggressive flight profile, using some combination of high speed and nap-of-the-earth (NOE) flight to achieve stealth, avoid attacks, and provide support to rapidly moving military operations. To date, unmanned rotorcraft have mostly been constrained to flight at altitudes well above ground-based obstacles, limited by the ability of the onboard controller or a remote pilot to sense and avoid. Unfortunately, the major benefit of rotorcraft lies in their ability to fly low and slow in the midst of an obstacle-rich environment. In order for unmanned rotorcraft to reach their potential, new methods for guidance, navigation, and control must be created that include the effect of a complex array of obstacles that are dynamically changing.

Fortunately, there is a rich literature emerging in the area of guidance and control techniques for air vehicle obstacle avoidance, including rotary-wing air vehicles. There are generally three categories of algorithms: geometric, optimization-based, and potential-field based. An early example of a geometric approach is from Cheng and Sridhar [1]. They found a gap in the obstacle field closest to the direction of travel that is wide enough for the aircraft to traverse. Frew and Sengupta [2] approached the problem by calculating “safe” and “unsafe” sets with layered control scheme. The method is extended to include the error model for camera-based vision and formations of multiple aircraft. Geometric approaches typically require low computational cost but sacrifice a guaranteed flight trajectory solution.

The second category of algorithms select a trajectory which optimizes a constrained cost function, usually some combination of obstacle distance, path length, and control input. An early approach by Gilbert explored the use of distance functions in the optimal control problem [3]. Lapp and Singh used a model predictive controller that optimizes a cost function over a finite prediction horizon, subject to hard constraints corresponding to obstacles [4 – 6]. Moon and Prasad [7] demonstrate a 2-D optimal control approach with hard constraints for both the obstacles and specific aircraft dynamic constraints, such as flapping angle and power.

One of the most common approaches to the problem is to model the operating area as a potential field. Early research on these methods was centered in the ground vehicle robotic community bu has expanded to many application areas, including air vehicles [8] [9] [10] [11] [12] [13]. The research has progressed to address basic problems, such as local minima, and apply the technique to practical problems [14] [15] [16]. A good example for rotorcraft is the work of, Scherer, et al [17] who demonstrated navigation of unmanned rotorcraft at high speeds and low altitudes among obstacles using a global path planning algorithm.

The work reported here employs a potential field method for autonomous rotorcraft navigation in an obstacle rich environment. While potential theory methods for vehicle navigation are by no means new, the work reported here proposes a set of modifications to uniquely tailor the algorithm to autonomous rotorcraft navigation. A technique to allow both velocity and acceleration constraints to be satisfied by shaping the speed profile along a spatial path is detailed. A mission objective dial is

† Graduate Research Assistant
* Sikorsky Associate Professor, Member AHS

defined which enables the operator to smoothly vary the computed path from one extreme of maximum aircraft masking to another extreme of maximum speed to the end destination. The characteristics of the proposed algorithm are shown through a set of simulated missions in realistic urban obstacle fields using practical sensor suite configurations. The paper begins with a discussion of the mathematical representation of the aircraft, which is followed by a description of the obstacle field mapping solution along with the associated path generation scheme. Example results are subsequently presented followed by a discussion of the results and conclusions.

Aircraft Dynamic Model

The work reported here is predominantly concerned with outer loop control of autonomous rotorcraft in an an obstacle field. It is assumed that an autopilot allows an aircraft to respond to velocity commands with a first order lag [18] (see Figures 1 and 2). Therefore, in formulating this problem the aircraft is modeled as a particle responding in the same way, with the standard state-space representation of the system as given in equations (1) through (3).

$$\dot{X} = A\,X + B\,U \tag{1}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_z} \end{bmatrix} \tag{2}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{\tau_x} & 0 & 0 \\ 0 & \frac{1}{\tau_y} & 0 \\ 0 & 0 & \frac{1}{\tau_z} \end{bmatrix} \tag{3}$$

In Equation 1, the state vector contains the position and velocity of the mass center of the aircraft $X = [x, y, z, v_x, v_y, v_z]^T$ while the control vector contains velocity commands $U = [v_{xc}, v_{yc}, v_{zc}]^T$. In equations 2 and 3, each τ represents the time constant corresponding to the lag response in the each direction.

A model predictive controller (MPC) is constructed to minimize the error between predicted trajectory and commanded trajectory for this work. However, any tracking flight control system can be used.

Path Planning

A critical part of obstacle field navigation is the ability to dynamically locate obstacles in the local area around the aircraft. Moreover, as obstacles are detected a map of obstacles in a three-dimensional environment must be constructed. This task is accomplished by defining a 3D grid around the aircraft as a map of the surrounding terrain features. The airspace around the aircraft is discretized and mapped to the array. Each element of the array is binary: 1 representing an occupied space, and 0 representing an empty space. Figures 3 and 4 represent a two-dimensional example of this type of grid, called an occupancy grid. Each set of sensor measurements are used to update this array. This approach has the advantage that the size of the obstacle map is independent of sensor type and the number of sensor readings accumulated. Also, redundant sensor readings do not take extra space. This is a simplified version of the evidence grid technique shown in [17].

The purpose of the path planner is to find a smooth, continuous obstacle-free path from the aircraft's current location to a desired waypoint. The mathematical machinery of potential theory provides a means to this end. In particular, the velocity field of an inviscid fluid flow around a body in the study of aerodynamics holds these characteristics, and can be represented as the gradient of a scalar potential function, ϕ.

$$V = \nabla \phi \tag{4}$$

Generally speaking, artificial potential field techniques formulate this problem by representing the goal point and obstacles as known spatial boundary conditions. The goal point Dirchlet condition on the potential function is set at -1 and the obstacles and space boundaries at 0.

The continuity equation, $\nabla \cdot V = 0$, reduces equation (4) to Laplace's equation:

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \tag{5}$$

A finite difference approximation is applied to Laplace's equation to form a discrete potential field algebraic equation:

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\phi_{i+1,j,k} - 2\,\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} \tag{6}$$

Substituting (6) and similar approximations in the y and z directions into (5), assuming an evenly spaced grid $(\Delta x = \Delta y = \Delta z)$, and solving for $\phi_{i,j,k}$ obtains

$$\phi_{i,j,k} = \frac{\phi_{i+1,j,k} + \phi_{i-1,j,k} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{i,j,k+1} + \phi_{i,j,k-1}}{6} \tag{7}$$

In other words, the value of each point in the discrete potential field is equal to the average of the six points around it.

Once the array specifying ϕ is found, then the same finite difference approximation can be used to calculate the gradient vector at each discrete point in the field. The streamline is then calculated from the vehicle's starting position by 4th order Runge-Kutta integration of the gradient vector field using linear interpolation. The trajectory follows the gradient to the point of lowest potential, the goal. Note that this algorithm only produces a path in space and does not address the speed at which to fly.

The solution to this boundary value problem requires a starting guess and an iterative process. As each cell is updated, its new value is in turn used to update subsequent cells. The order of update is alternated from iteration to iteration in order to prevent a bias from entering the potential field. The number of iterations required to converge depends upon the size of the array, the quality of the starting guess, and the convergence criteria used to terminate the algorithm. The algorithm can be significantly sped by using techniques detailed by Scherer, et al [17], including multi-grid, iterating only until the solution has no local minima, and setting the starting guess as a previous solution to the obstacle field.

An example two-dimensional potential field with two-dimensional obstacles is shown in Figure 5.

Speed Shaping

As stated above, the artificial potential field method provides only a path through the obstacle field but no details how to fly it.

Given a general twisting and turning obstacle free path, movement along this path at a constant velocity will cause changes in acceleration due to path curvature. In determining the speed to fly a particular path, maximum speed and acceleration limits must be satisfied. These limits may be basic aircraft limits, limits fed back to the algorithm from the inner-loop flight controller, or limits imposed by an operator based on a given mission scenario. It may be desirable to traverse a commanded trajectory slowly for a given mission (for example, overwatch of a ground-based element) while very rapidly for another (for example, solo reconnaissance.) Here, it is assumed that the aircraft's dynamic constraints, such as maximum flapping angle, power output, etc. can be mapped to a maximum velocity and a maximum acceleration of the vehicle. These values are known prior to start of flight or fed to the algorithm by the underlying flight controller.

Given the geometric path, the speed shaping algorithm seeks to find a speed profile that traverses the path in the shortest time without violating dynamic constraints. The first step is to parameterize each coordinate by pathlength, s (equations 8 and 9),

$$\Delta s_i = \left\| \vec{r}_i - \vec{r}_{i-1} \right\| \tag{8}$$

$$s_i = s_{i-1} + \Delta s_i \tag{9}$$

where $s_0 = 0$. The unit tangent and unit normal vectors to the curve as well as the curvature are found as a function of s using finite difference approximations (10)-(12).

$$\vec{t}_i = \frac{\frac{\vec{r}_{i+1} - \vec{r}_{i-1}}{2 \cdot \Delta s_i}}{\left\| \frac{\vec{r}_{i+1} - \vec{r}_{i-1}}{2 \cdot \Delta s_i} \right\|} \tag{10}$$

$$\vec{n}_i = \rho \left(\frac{\vec{t}_{i+1} - \vec{t}_{i-1}}{2 \cdot \Delta s_i} \right) \tag{11}$$

$$\rho = \frac{1}{\left\| \frac{\vec{t}_{i+1} - \vec{t}_{i-1}}{2 \cdot \Delta s_i} \right\|} \tag{12}$$

To compute the speed profile, the algorithm makes an initial guess for speed as a function of pathlength, typically just greater than zero to ensure that initial guess does not violate any dynamic constraint. Using that initial guess, time is found as a function of pathlength (13)-(15).

$$\Delta s_i = \frac{v_i + v_{i-1}}{2} \cdot \Delta t_i \tag{13}$$

$$\Delta t_i = 2 \cdot \left(\frac{1}{v_i + v_{i-1}} \right) \cdot \Delta s_i \tag{14}$$

$$t_i = \sum_{j=1}^{i} \Delta t_j \tag{15}$$

Finally, the acceleration over the curve is found (16).

$$\vec{a}_i = \frac{\Delta v_i}{\Delta t_i} \cdot \vec{t}_i + \frac{v_i^2}{\rho_i} \cdot \vec{n}_i + \vec{g} \tag{16}$$

The guess values for v(s) are considered point by point using the following logic: Is the point fixed? If yes, do nothing. Does the point meet or exceed specified constraint conditions (i.e. end points)? If yes, slightly reduce and fix the velocity at that point and move on to the next point. Does the point meet or exceed the overall velocity limit? If yes slightly reduce and fix the velocity at that point and move on to the next point. Does $\left\| \vec{a}_i \right\|$ or $\left\| \vec{a}_{i-1} \right\|$ exceed acceleration limits? If yes slightly reduce and fix the

velocity at that point and move on to the next point. If the answers to the previous four questions are no, then increment the velocity of the point upward and move to the next point.

Once all the points on the velocity profile have been fixed, then every point in the velocity profile has met a constraint and the optimum has been found. Figure 6 and 7 show an example solution.

Since this method produces a set of reference positions, velocities, and accelerations and a corresponding time for each, the results can be interpolated to find a sequence of commands to fly the entire route.

Vehicle Masking

An operator supervising the aircraft requires a straightforward way to balance the competing desires to fly fast and to fly low based upon mission requirements. A straightforward approach to this problem is to impose phantom obstacles in the form of an artificial ceiling and floor into the potential field. With the artificial ceiling or floor, it is possible to cut off all paths from start point to goal (see Figure 8). The solution to this problem is to create a "blanket" region around obstacles which overrides the imposed ceiling or floor. The aircraft is always left with a path over any obstacle if no lateral path exists.

The masking factor (κ) is set by the operator with a value between 0 to 1, 0 being no masking and 1 being maximum masking. From this parameter, a ceiling height is selected using equation (17), where h_C is the ceiling height and h_G is the average height of the ground. Note that because of the boundary conditions in the path planning algorithm, there is always a ceiling at the top of the flight area, here denoted at h_{FA}.

$$h_C = h_{FA} - \kappa(h_{FA} - h_G) \qquad (17)$$

The blanket area is calculated by starting with the occupancy grid and propagating the occupancy grid one grid square/cube at a time until sufficient clearance has been achieved. The blanket must extend out at least twice the desired standoff distance from an obstacle, since the streamline will be halfway between the ceiling and the obstacle.

Finally, the blanketed volume is subtracted from the ceiling to produce a modified ceiling. This ceiling is added to the obstacle map and is otherwise treated as an obstacle. See figures 9 through 10 for example.

Note that masking is inversely related to speed; a masked path will tend to produce sharper bends in the planned path, and the velocity planner accordingly slows the aircraft to accomplish such turns.

While the potential theory based path planning algorithm produces smooth trajectories for a given set of boundary conditions, there is no guarantee that the new path is smoothly joined to the previous one when new boundary conditions are imposed based on new information. In fact, at times very sharp cusps can be formed (see Figure 11). Normally this happens when a sufficiently large obstacle comes into sensor range which the path planner had previously determined to fly through. To remedy this situation, the potential field is modified directly allowing smooth transition from one computed path to another. This option is called the "chase" boundary condition. The chase condition is defined so that a phantom obstacle chases the aircraft through the obstacle field. If an obstacle appears which would normally cause the aircraft to reverse direction abruptly, the chase obstacle provides a counteracting force in the potential field (see Figures 12 and 13). The resultant trajectory, while still not necessarily smooth, is at a much shallower angle.

Results

To explore the capabilities of the above path planning algorithm, a set of simulation results have been generated for an example aircraft responding to velocity commands with a first order lag. The aircraft which the simulation approximates is the GTMax, a testbed aircraft at Georgia Tech (see Figure 15). This aircraft has a 10.2 ft rotor diameter, 205 lb gross weight, a 21 hp powerplant, and 2 onboard computers which operate autopilot. In the lateral directions, the aircraft reacts to velocity commands with a time constant of 0.8 seconds, and in the vertical direction with a time constant of 1.2 seconds (see equations 2 and 3).

First, a set of cases are shown where the path planner has perfect or near-perfect information about the obstacle field. The masking factor is varied from 0 to 0.9. The terrain is loosely based upon McKenna Urban Training Site at Ft. Benning, Georgia, USA (Figure 16), with additions and variations to show the planning algorithm's avoidance capabilties. Note that the planned path for no masking follows a benign curve where the aircraft initially climbs, then follows a relatively straight path over the top of the obstacle field before descending to the waypoint. In each successively greater case of masking, the path is closer to the ground and much curvier (Figures 17 through 20). Interestingly, in the case of greatest masking, the planner selects an entirely different route through the obstacle field.

Further study included the simulation of the aircraft gathering sensor data, then updating the path planned. For this work a notional two-axis scanning laser rangefinder is used. The sensor has sufficient mobility that it is capable of aligning itself with the aircraft's velocity vector (Figure 13). During a single laser scan a set of pulses is fired at different azimuths and radii. For each pulse fired, a return is generated that provides an estimate of the range to an obstacle.

This sensor is modeled in simulation by representing the obstacle field as a group of planes, which are stored as a rotation matrix (T_{PI}), a corner position, and a length and

width. During simulation, the virtual sensor solves a vector algebra problem, as shown in (18), for the distance from the sensor to the point where the laser strikes a plane. Error is placed on the distance reading according to equation (19). The inertial position of the point of strike is then solved for using equation (20). This process is repeated for all planes in the obstacle field and across of the sensor's field of view (see Figure 14).

$$\begin{gathered}\mathbb{C}_I[\vec{r}_{H\to P}]+\mathbb{C}_I[\vec{r}_{P\to A}]=\mathbb{C}_I[\vec{r}_{H\to A}]\\ \mathbb{C}_H[\vec{r}_{H\to A}]=d\cdot\mathbb{C}_I[\vec{n}_{lidar}]\end{gathered} \qquad (18)$$

$$d_{actual}=\frac{\begin{bmatrix}1 & 0 & 0\end{bmatrix}\cdot T_{PI}^T\cdot(\vec{r}_{O\to P}-\vec{r}_{O\to H})}{\begin{bmatrix}1 & 0 & 0\end{bmatrix}\cdot T_{PI}^T T_{HI}\mathbb{C}_H[\vec{n}_{lidar}]}$$

$$d_{measured}=d_{actual}\cdot(1+\sigma\cdot w)+b \qquad (19)$$

$$\vec{r}_{O\to A}=\vec{r}_{O\to H}+d_{measured}\cdot\mathbb{C}_I[\vec{n}_{lidar}] \qquad (20)$$

For the results shown, the laser rangefinder has a maximum range of 80 meters resolved to 1 meter and field of view of 160 degrees, resolved to roughly 1 degree. The laser was simulated with no noise or bias.

The tracking controller used is a standard model predictive controller with a prediction horizon of 3 seconds and equal weighting on tracking error and control input terms. The controller generates velocity commands at 10 Hz.

In order to quantitatively compare flights, a metric of some kind is required. There are three possible metrics which seem to be the most straightforward way to measure how fast and how low the aircraft flies: time of flight, the maximum altitude of flight, and the average altitude over the time of flight (which roughly measures the amount of time exposed, Equation 21).

$$\sum_i z_i\cdot\Delta t_i \qquad (21)$$

To generate the following results, the path planner and obstacle map were composed of 128x128x64 arrays, with each cell in the array representing a 2 m x 2 m x 1 m space.

Three sets of case studies are examined. First, the change in performance while varying masking factor from 0 to 1 is examined. Second and third, maximum velocity and maximum acceleration are examined.

Case Study 1: Masking

The first case study provided the most interesting results. In this study, six simulations were run with differing masking factors, from 0 to 1 in increments of 0.2 (see Table 1 and Figures 21 and 22). For up to 60% masking (approximately a 25 m ceiling), the aircraft actually flew a little faster than with no masking at all, in addition to maintaining a far lower profile. With no masking, the space of lowest potential is situated roughly halfway between the ground and the top of the space, so that the aircraft will initially climb before accelerating in the horizontal. Two conclusions may be drawn. First, for optimal performance, the base ceiling may need to be set to about 50% of its current value or the aircraft may need to use a finer potential grid in the vertical direction. Second, the course in simulation was too short to appreciate the full range of performance. In a longer flight, the aircraft can fly faster at an altitude of 35 m than it could at 15 or 20 m because the higher altitude would be far freer of obstruction and allow the aircraft to fly in a straight line a full speed.

Table 1. Max Velocity = 70; Max Acceleration = 4g; Chase Obstacle On

Masking Factor	Time of Flight	Max. Altitude	$\sum_i z_i\cdot\Delta t_i$
0	10	36	244
0.2	10.2	35	243
0.4	9.7	25.8	182
0.6	9.4	21.6	151
0.8	10	17.6	138
1	>20	12	Did not reach goal in simulation time

Case Studies 2 and 3: Velocity and Acceleration Constraints

The range of velocity constraints and acceleration constraints also have two notable features. First, in this simulation set, top speed was limited primarily by the aircraft's ability to accelerate and decelerate. Even with a maximum velocity constraint of 80 m/s, the aircraft didn't get going much faster than 50 m/s. The second notable feature is that an increase in acceleration constraint marginally improves the ability to fly at low altitudes due to increased capability to execute sharp turns.

Table 2. Masking Factor = 0.5; Max Acceleration = 4g; Chase Obstacle On

Max Velocity (m/s)	Time of Flight	Max. Altitude	$\sum_i z_i\cdot\Delta t_i$
30	11.6	22.35	200
40	10	21.9	165
50	9.6	21.7	155
60	9.4	21.6	151
70	9.4	21.6	151
80	9.4	21.6	151

Table 3. Masking Factor = 0.5; Max Velocity = 70; Chase Obstacle On			
Max Acceleration (m/s^2)	Time of Flight	Max. Altitude	$\sum_i z_i \cdot \Delta t_i$
2g	13.5	22.25	221
2.5g	11.7	22.04	191
3g	10.8	22.02	176
3.5g	9.9	21.8	160
4g	9.4	21.6	151
4.5g	9.1	21.6	146

Conclusions

A potential-field algorithms for obstacle field navigation was demonstrated to plan safe paths to a goal point using mission criteria to determined altitude and speed of flight. The masking function was shown to work well to reduce the aircraft's exposure to threat with little or no penalty in performance over short distances. The adjustments to velocity and acceleration constraints offer increases in performance, but only to a point.

References

1. Cheng VH, Sridhar B. “Considerations for Automated Nap-of-the-Earth Rotorcraft Flight.” 1988 American Control Conference.
2. E. Frew, R. Sengupta, “Obstacle Avoidance with Sensor Uncertainty for Small Unmanned Aircraft ,” 43rd IEEE Conference on Decision and Control (2004).
3. Gilbert EG, Johnson DW. “Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles.” IEEE Journal of Robotics and Automation. 1985;RA-1(1):21-30.
4. T. Lapp, L. Singh, “Model Predictive Control Based Trajectory Optimization for Nap-of-the-Earth (NOE) Flight Including Obstacle Avoidance ,” Proceeding of the 2004 American Control Conference30 June-2 July 2004, Volume: 1, 891- 896.
5. T. Lapp, L. Singh, “Model predictive control in nap-of-Earth flight using polynomial control basis functions,” Proceeding of the 2005 American Control Conference, 8-10 June 2005, 840-845.
6. Singh L, Fuller J. “Trajectory generation for a UAV in urban terrain, using nonlinear MPC.” Proceedings of the 2001 American Control Conference. . 2001:2301-2308.
7. J. Moon, J.V.R. Prasad, “Minimum-time Approach to Obstacle Avoidance Constrained by Envelope Protection for Autonomous UAVs,” Proceedings of the 65th Annual Forum of the American Helicopter Society, Grapevine, TX, May 27 – 29, 2009.
8. Khatib, Oussama. 1986. “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots.” International Journal of Robotics Research. 5, no. 1: 90-98.
9. Krogh B, Thorpe C. “Integrated path planning and dynamic steering control for autonomous vehicles.” Proceedings. 1986 IEEE International Conference on Robotics and Automation.:1664-1669.
10. C. W. Warren, “Global Path Planning Using Artificial Potential Fields,” IEEE International Conference on Robotics and Automation, 1989.
11. C. W. Warren, “A Technique for Autonomous Underwater Vehicle Route Planning,” IEEE Journal of Oceanic Engineering, Vol. 15 Issue 3, 199-204, 1990.
12. C. W. Warren, “Multiple robot path coordination using artificial potential fields.” Proceedings., IEEE International Conference on Robotics and Automation. 1990:500-505.
13. Y. Koren and J. Borenstein, “Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation,” Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, 1991, pp. 1398-1404.
14. Z.X. Li and T.D. Bui, "Robot Path Planning Using Fluid Model," Journal of Intelligent and Robotic Systems, vol. 21, 1998, pp. 29-50.
15. M. Park, J. Jeon, M. Lee, “Obstacle Avoidance for Mobile Robots Using Artificial Potential Field Approach with Simulated Annealing,” IEEE International Symposium on Industrial Electronics, 2001.
16. S. Ge and Y. Cui, “Dynamic Motion Planning for Mobile Robots Using Potential Field Method,” Autonomous Robots, 2002.
17. S. Scherer, S. Singh, L. Chamberlain, M. Elgersma, “Flying Low and Fast Among Obstacles: Methodology and Experiments,” The International Journal of Robotics Research, Vol. 27, No. 5, 549-574 (2008).
18. J. Moon, R. Sattigeri, J.V.R. Prasad, and A. Calise, “Adaptive Guidance and Control for Autonomous Formation Flight,” Proceedings of the 63rd Annual Forum of the American Helicopter Society (2007).
19. W. Levine, ed. *The Control Handbook.* CRC Press LLC, 1996.

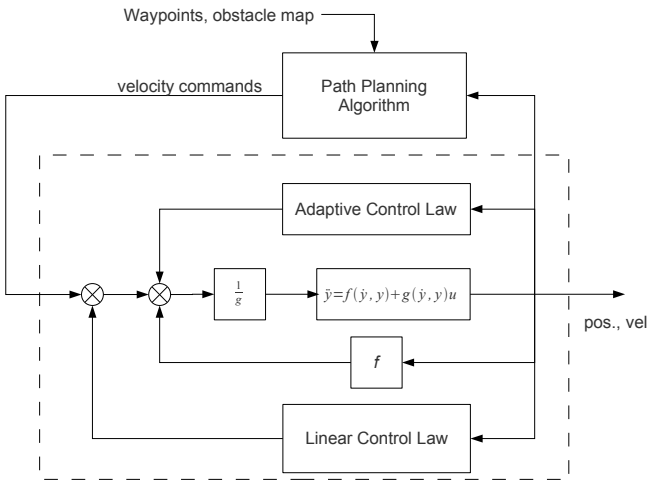

Figure 1. Autopilot Block Diagram

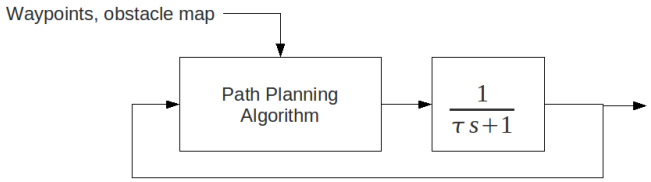

Figure 2. Dynamic Model Block Diagram

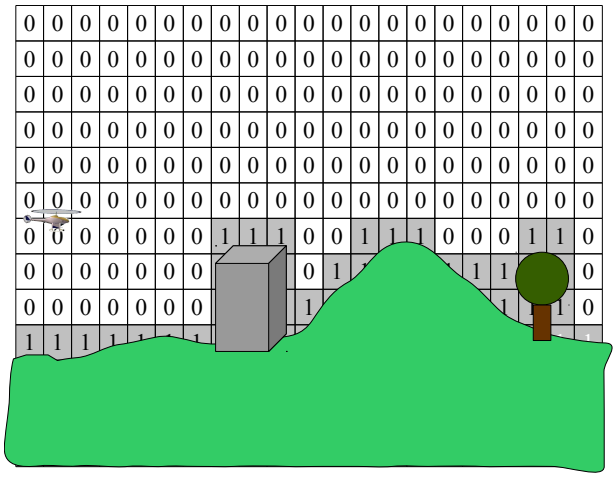

Figure 3. Terrain Superimposed on Occupancy Grid

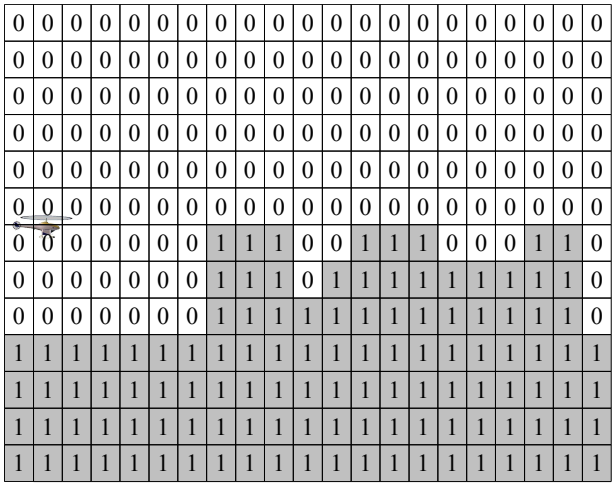

Figure 4. Resultant Occupancy Grid

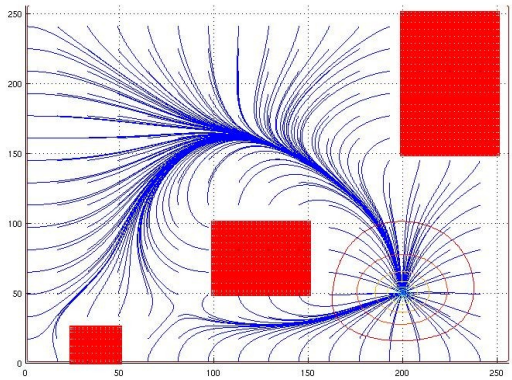

Figure 5. 2D Potential Field Example w. Streamlines

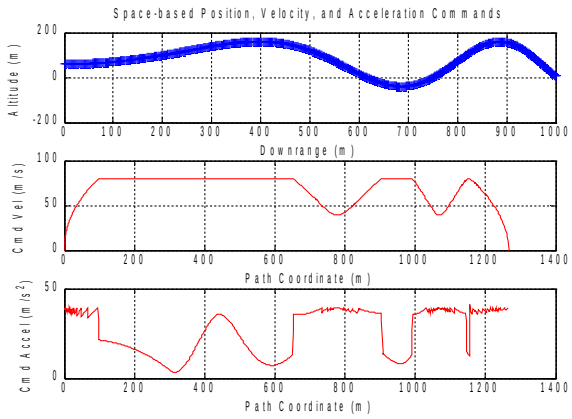

Figure 6. Speed Shaping Example, pathlength domain

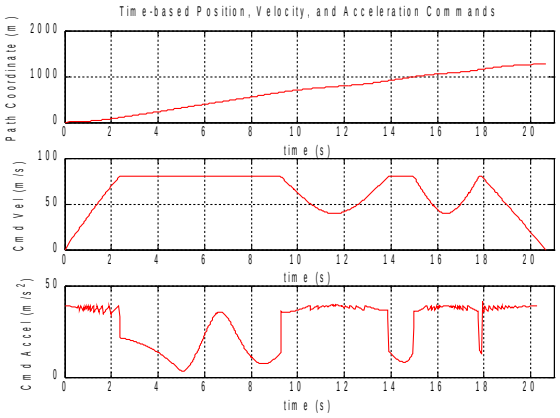

Figure 7. Speed Shaping Example, time domain

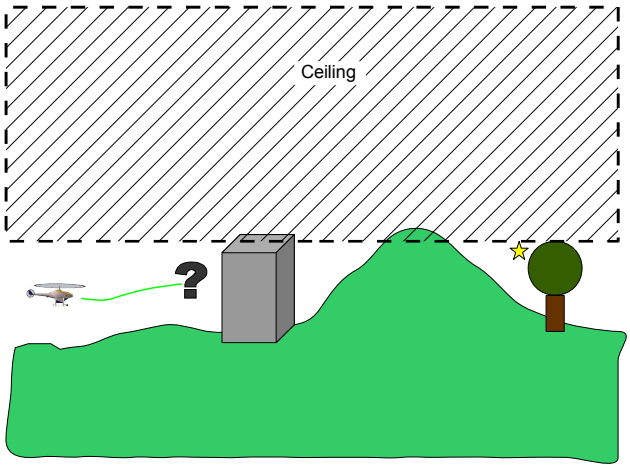

Figure 8. Simple Ceiling

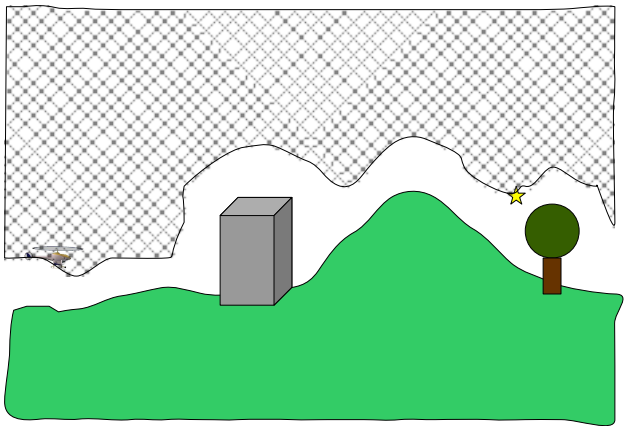

Figure 9. Fully Blanketed Terrain

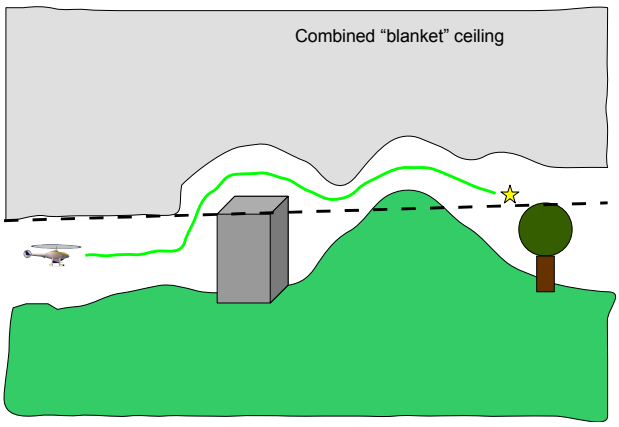

Figure 10. Modified Ceiling

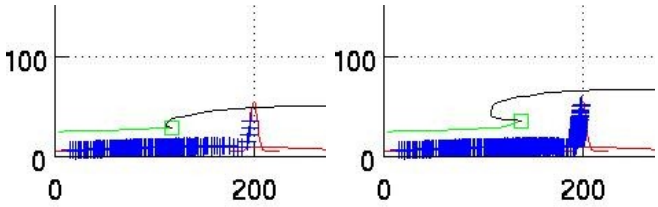

Figure 11. Sharp Cusp with Trajectory Update

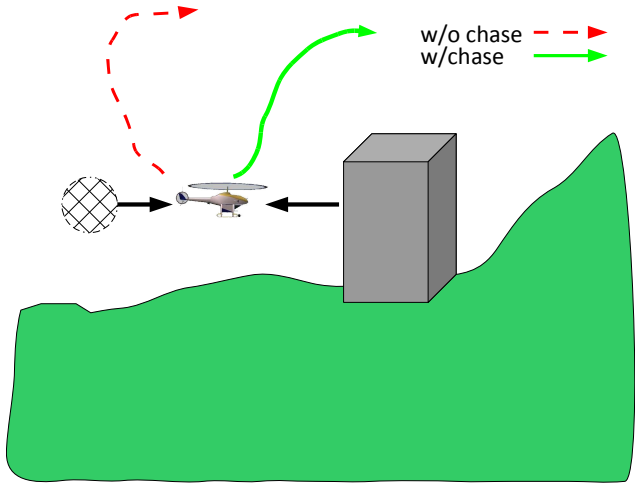

Figure 12. Illustration of Sharp Cusp and Counteractive Effect of Chase Obstacle

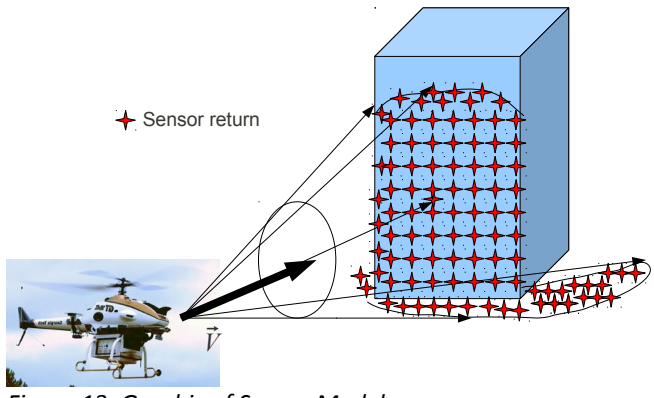

Figure 13. Graphic of Sensor Model

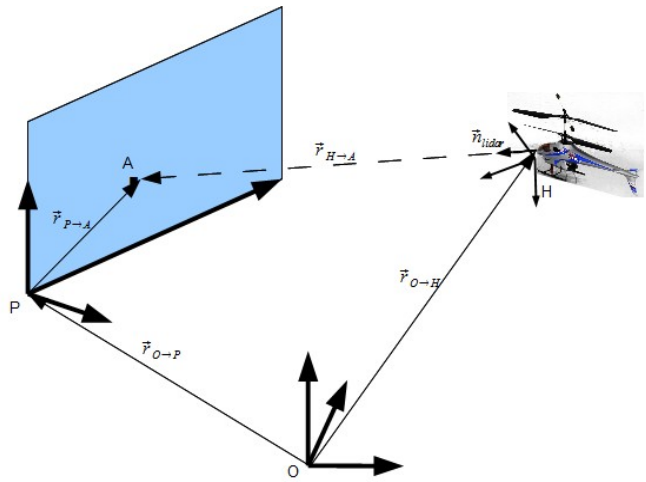

Figure 14. Virtual Sensor Geometry

Figure 15: GTMax Research Helicopter

Figure 16. McKenna Urban Training Site, Fort Benning, GA

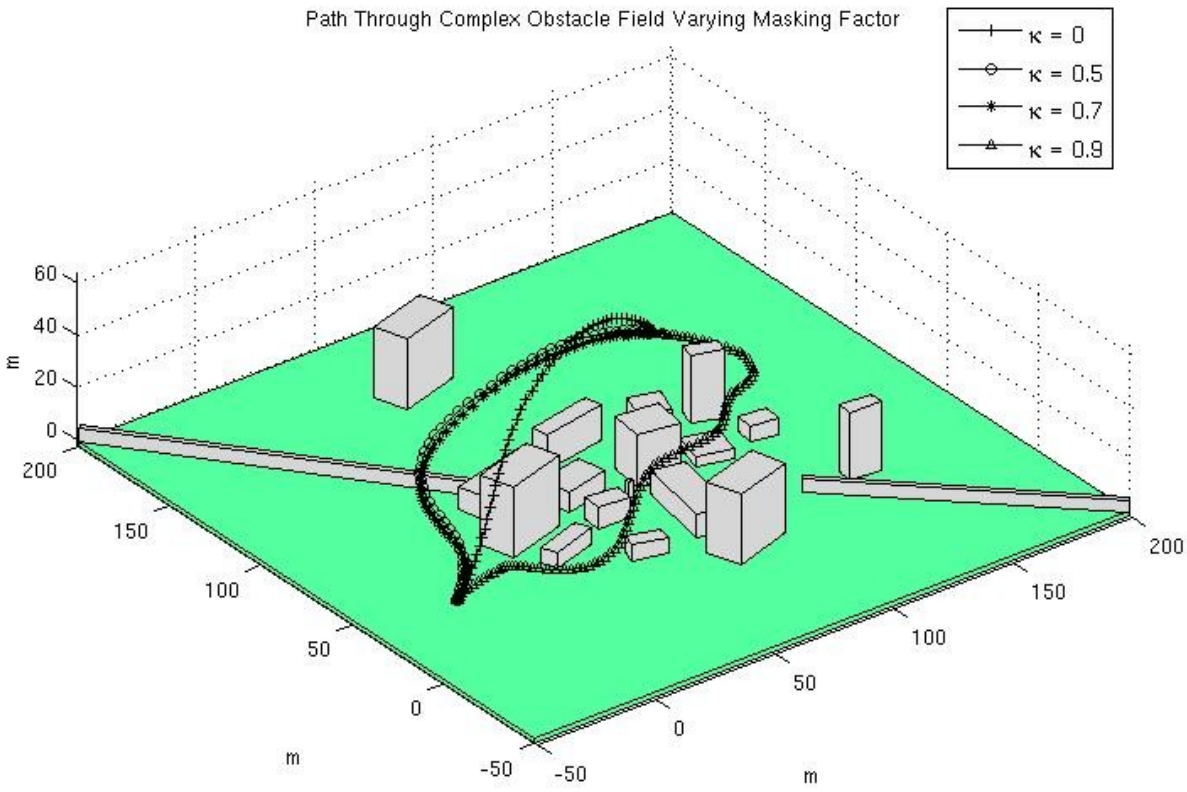

Figure 17. Paths Planned with Perfect Information, Oblique View

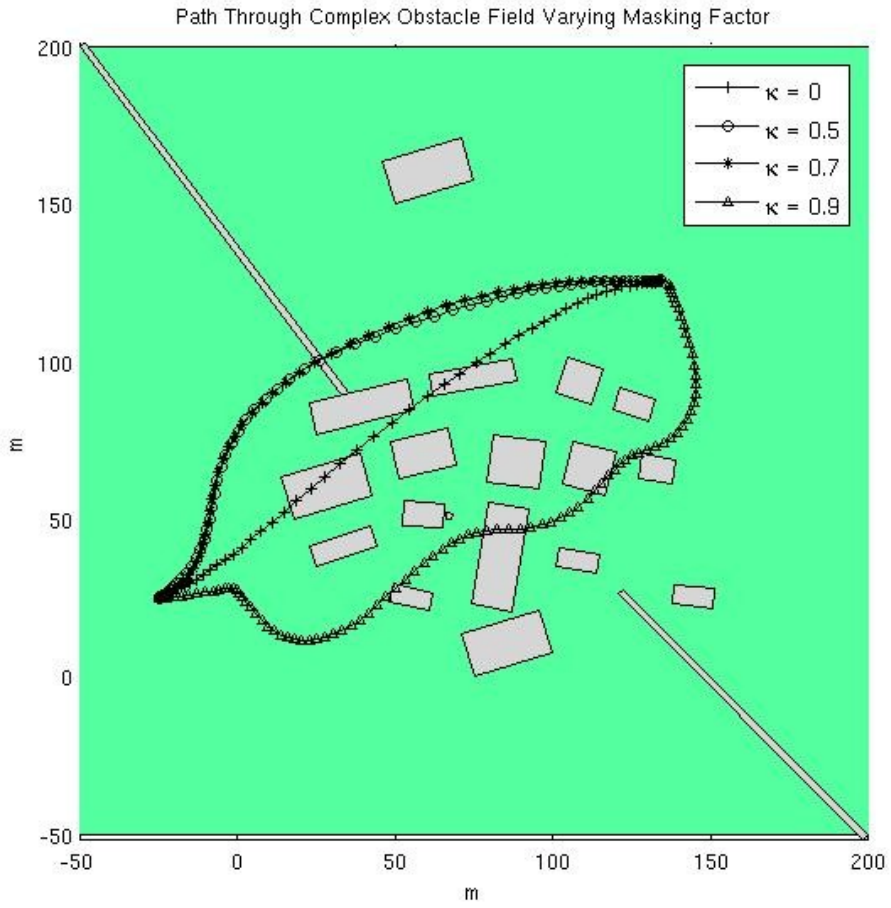

Figure 18. Paths Planned with Perfect Information, Top View

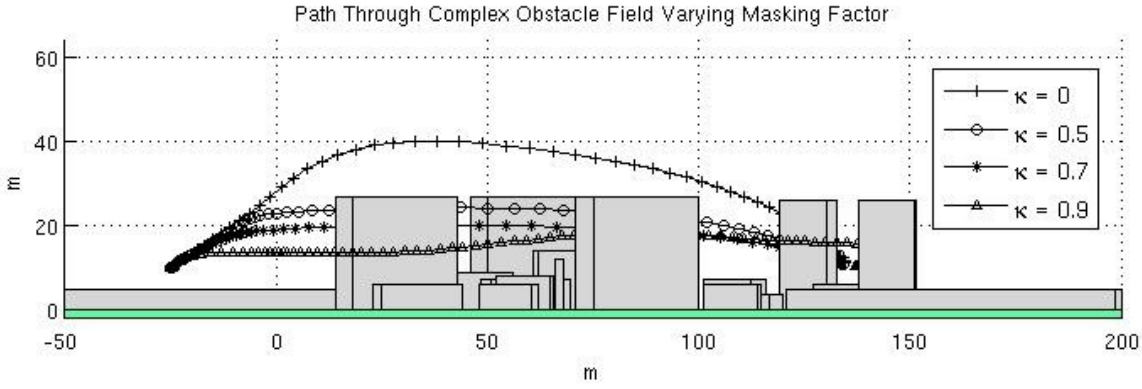

Figure 19. Paths Planned with Perfect Information, Y-Z Plane View

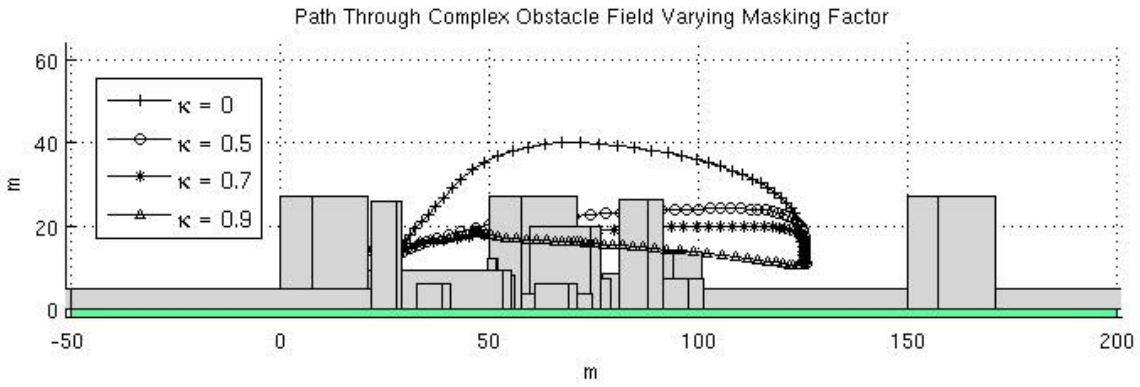

Figure 20. Paths Planned with Perfect Information, X-Z Plane View

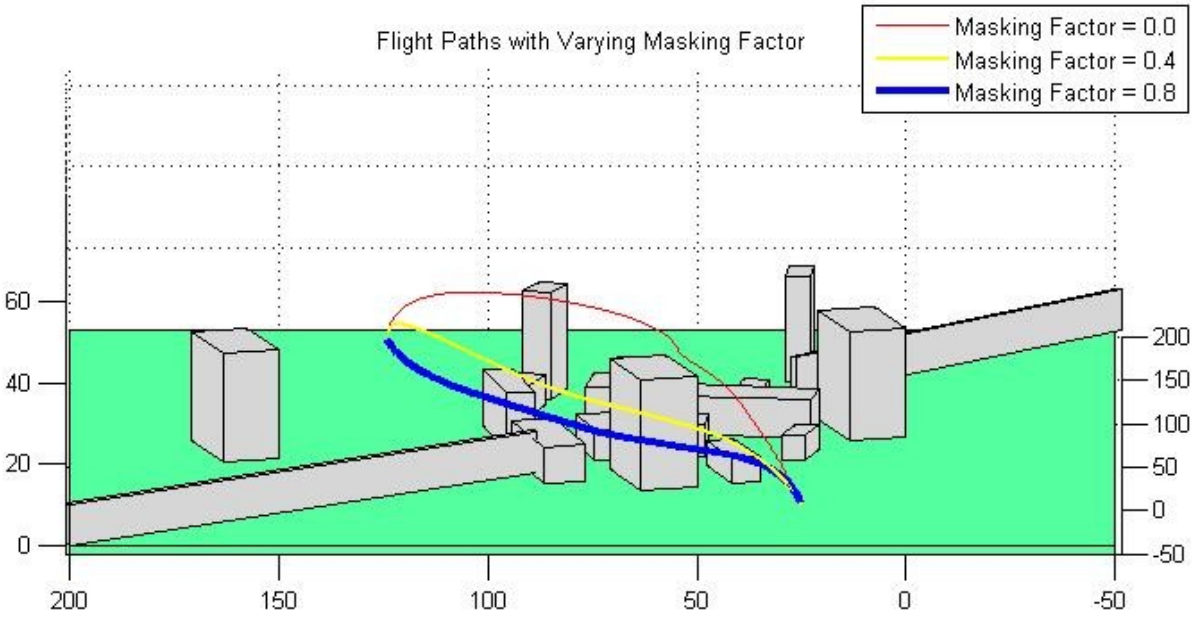

Figure 21. Simulated Trajectories, Oblique View

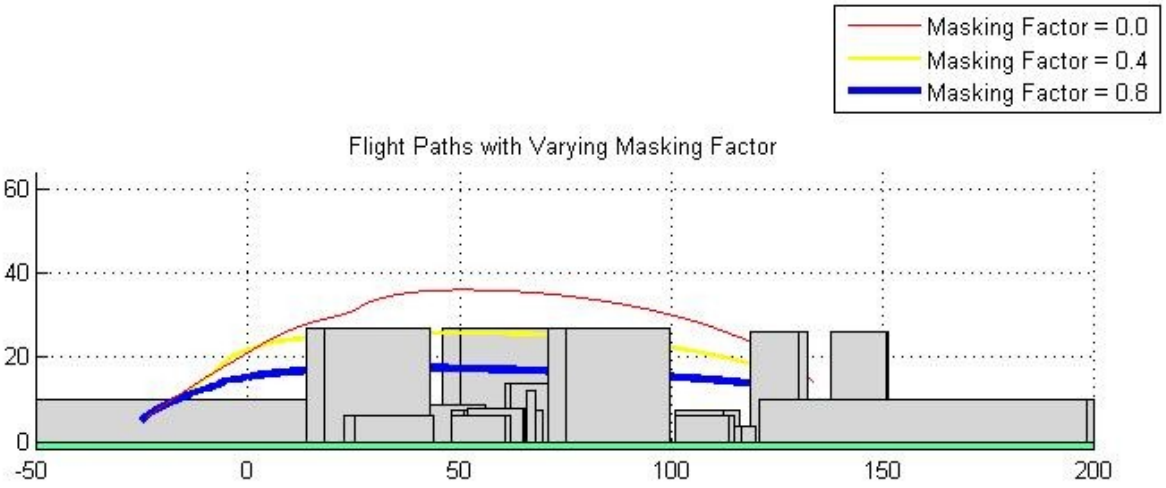

Figure 22. Simulated Trajectories, X-Z plane view